Graphics and Imaging Performance Tuning

Session 516

Graphics and Imaging Performance Tuning

Haroon Sheikh
Manager, Graphics Software

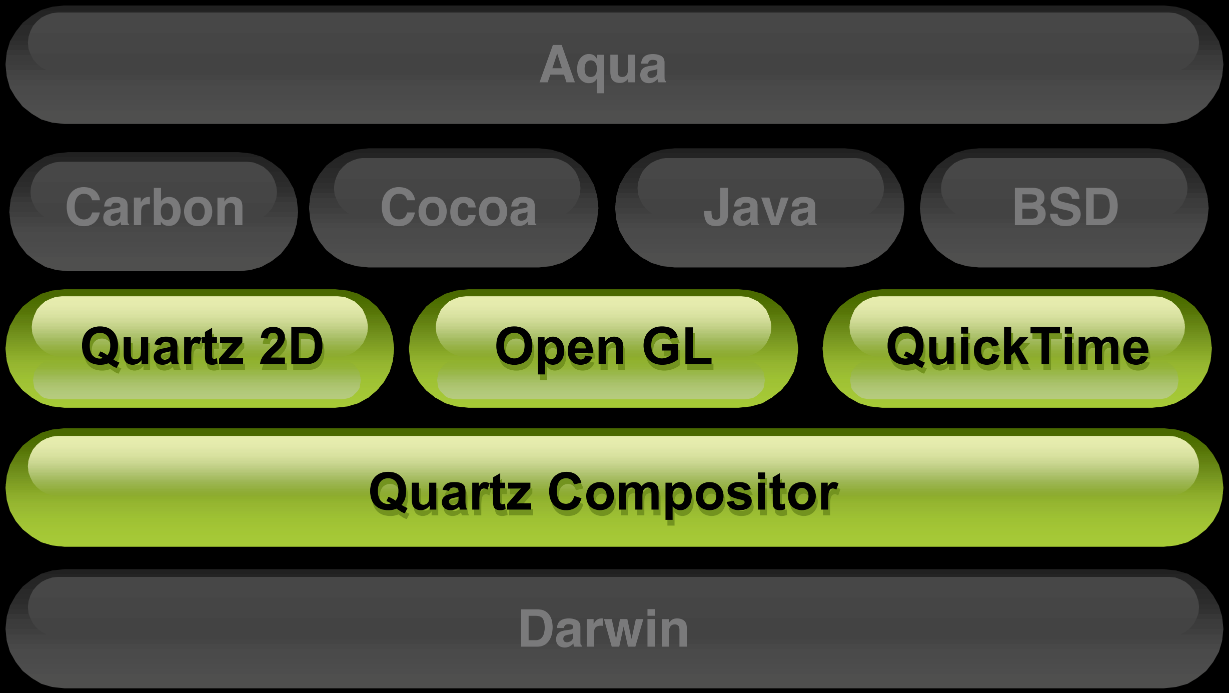
Topics

- Performance Tuning
  - Quartz Compositor
  - QuickDraw Performance
  - Demos
- Tips and recommendations
  - QuickDraw
  - Quartz 2D

Performance Opportunities

- System level optimizations
- Recommendations
- Your feedback is important!

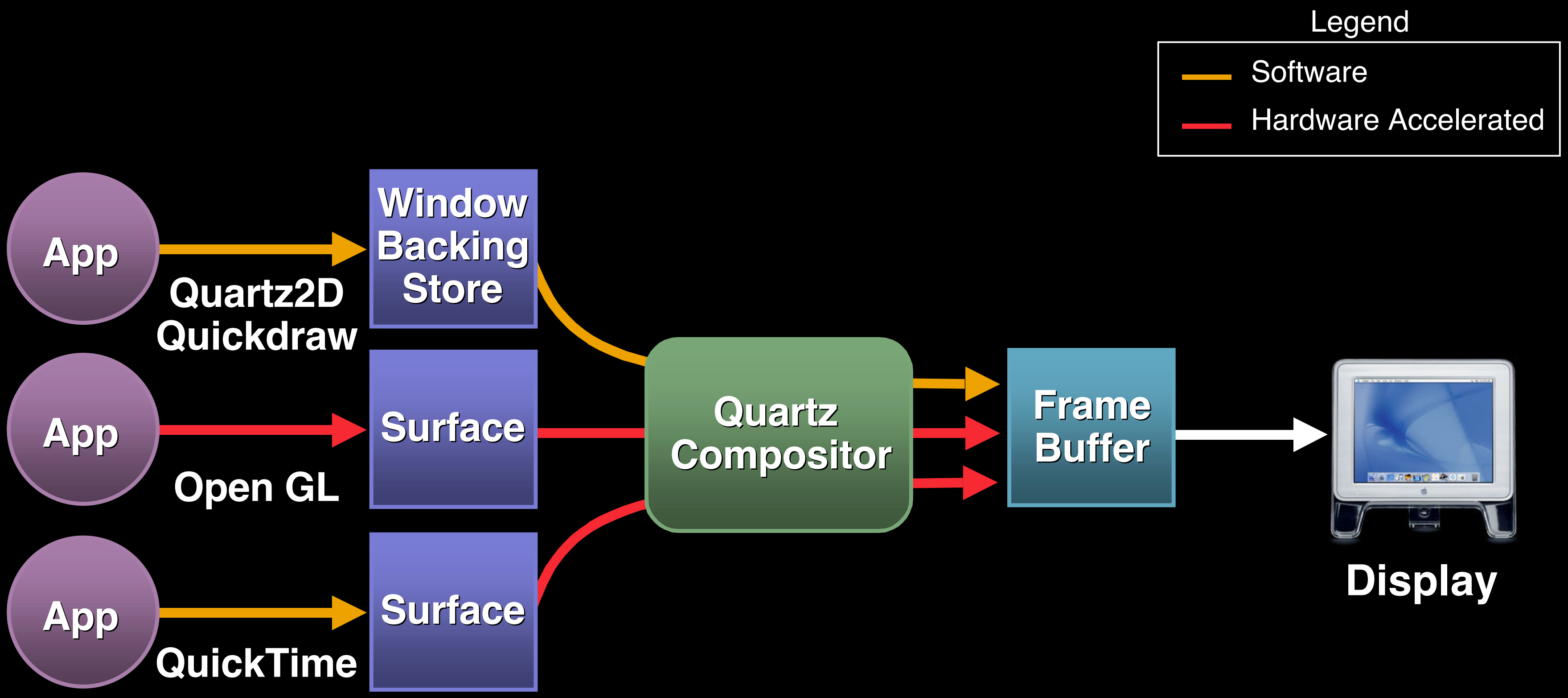
Architecture Diagram



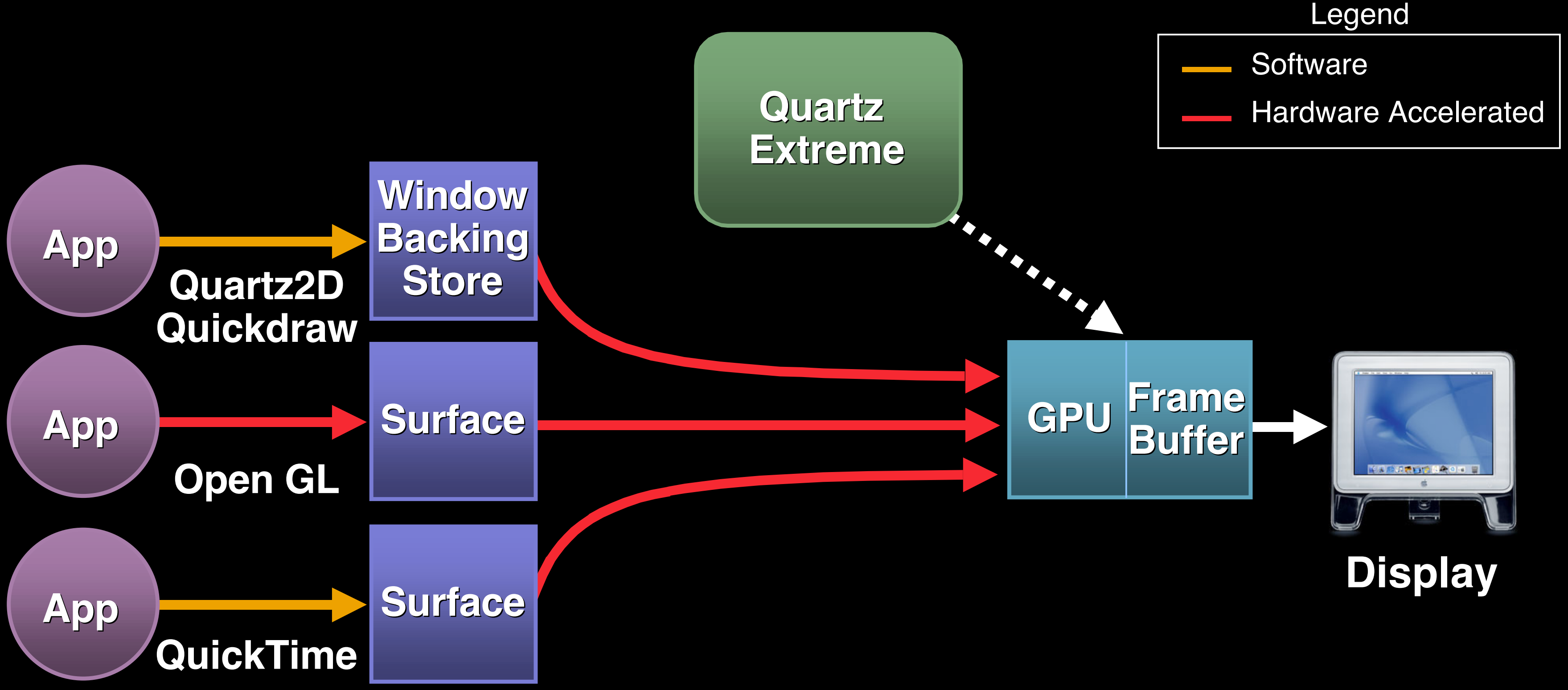
Quartz Compositor

- Compositor presents window content from multiple applications to the display
- Hardware acceleration is used to perform composite operation and flushing
- Accomplished through window back buffer abstraction

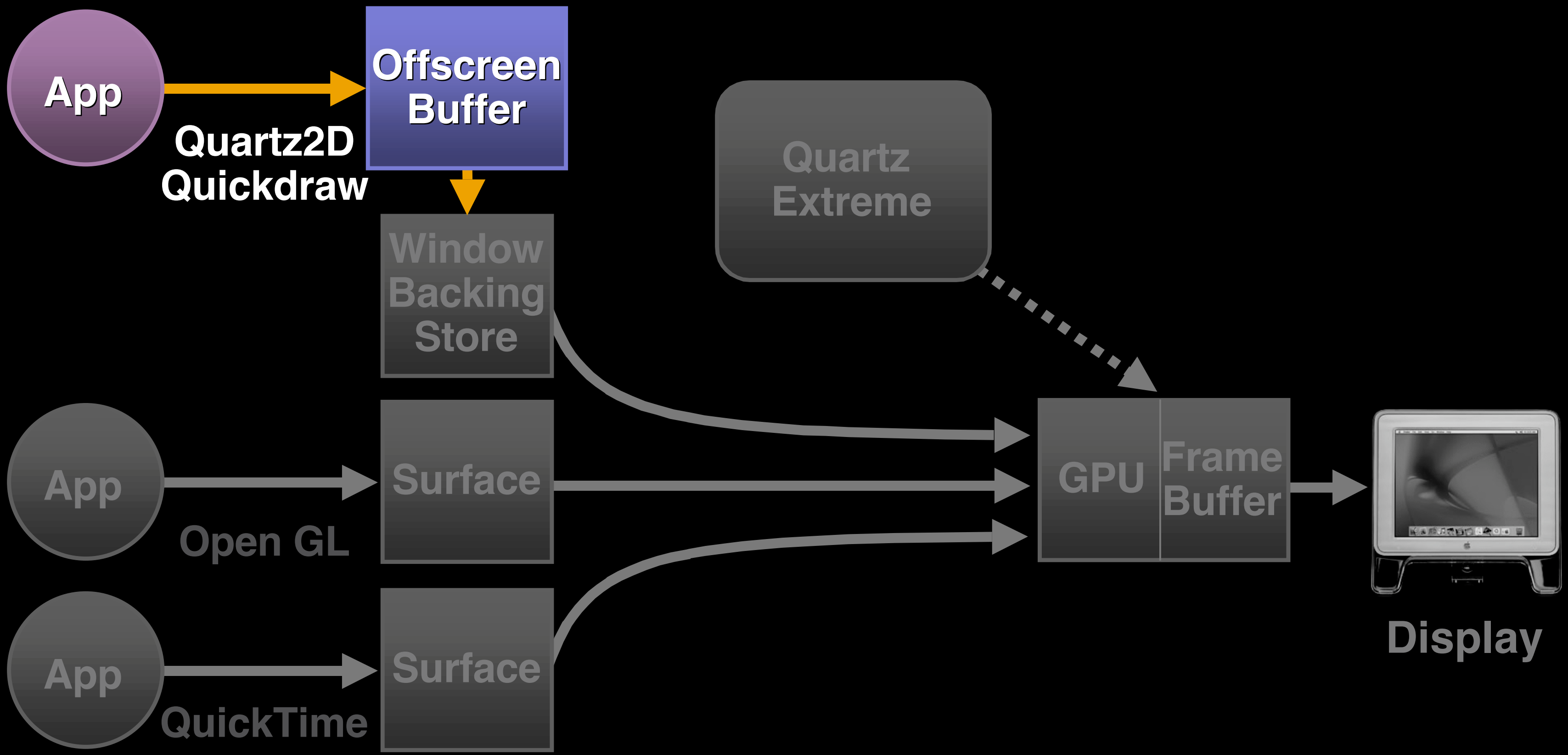
Software Quartz Compositor



Accelerated Quartz Compositor



Avoid extra buffer



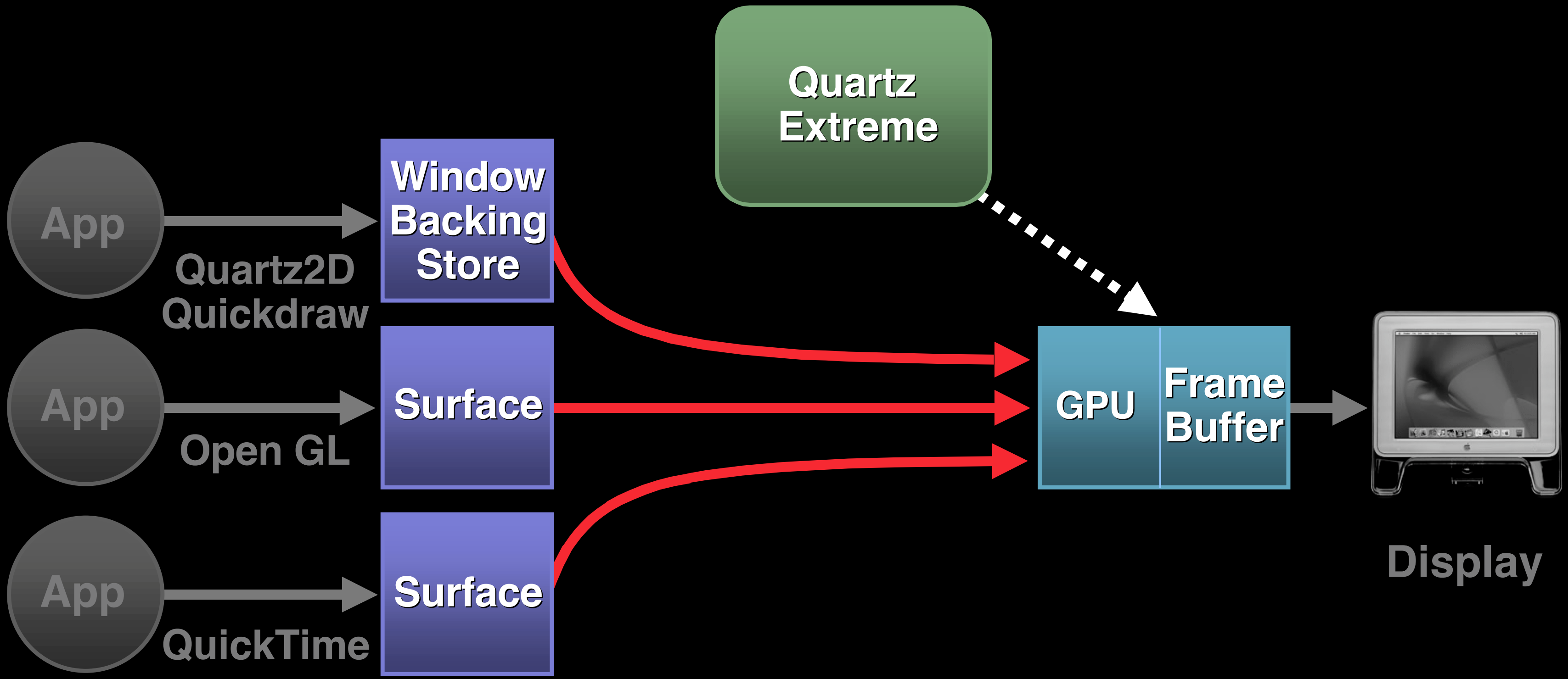
Recommendations

- Avoid unnecessary off-screen windows
  - Window back buffers consume memory
- Use one-shot windows
  - Window repaints are usually faster than paging in backing store memory
  - Carbon and Cocoa do this differently

Recommendations

- Avoid re-sizing immediately after window creation
  - Causes redundant operation
- Do not assume backing store will be in system memory
  - Need to maintain backing store abstraction

Flushing


Quartz Extreme
App
Quartz2D
Quickdraw
Window Backing Store
App
Open GL
Surface
App
QuickTime
Surface
GPU
Frame Buffer
Display

Flushing

- Implicit
  - Carbon and Cocoa
- Explicit

```
QDFlushPortBuffer
CGContextFlush
[NSWindow flushWindow]
```

Asynchronous Flushing

- Flush is asynchronous
- Subsequent drawing to window buffer will block if flush has not completed
- Tip: Fastest way to **frame buffer** is to draw into window backing store and flush
    - Quartz Extreme is even faster!
    - AGP transfer is used

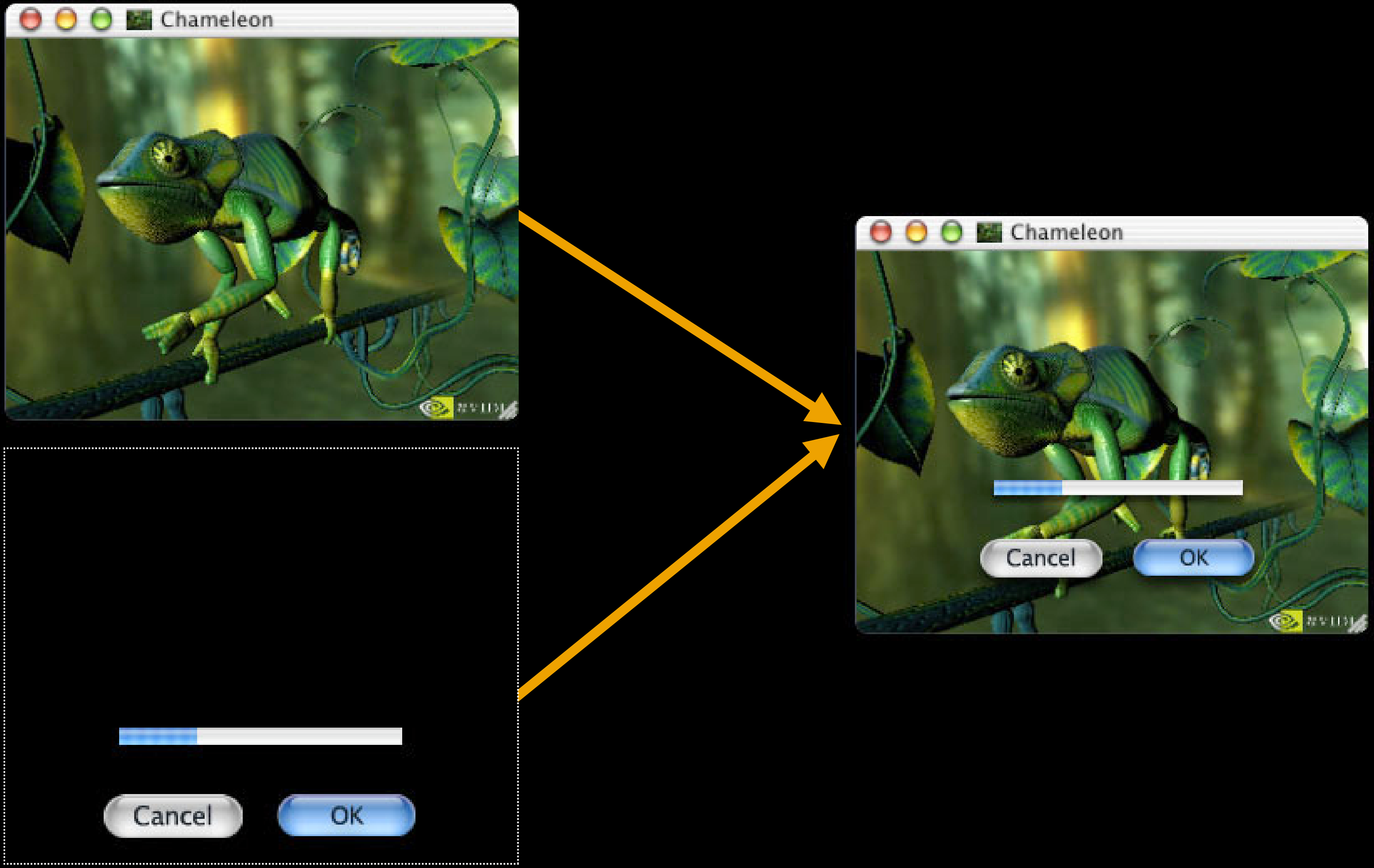
Beam Sync

- Flushing is synced to CRT beam to avoid tearing
- LCD panels also have a screen update frequency
- Recommendations:
  - Avoid redundant flushes
  - Aim for 30 to 60 fps

Scroll Acceleration

- Given the system memory to VRAM bottleneck, minimize the data transferred
- System level optimization:
    - At the lowest levels operation is up to 3x faster
    - **ScrollRect, NSScrollView** or CG imaging will improve by 30% depending on scroll area
    - Copying from and to the same window buffer also uses this optimization

Overlays



Overlay Acceleration

- System level optimization:
  - Quartz Extreme significantly improves performance of overlays
- Tip: Use Carbon Overlay Window class
  - **CGContextClearRect**
  - QuickDraw writes FF for alpha into window buffer

Backing Store Compression

- Window buffers are one of the major consumers of memory
- After a delay, inactive windows are compressed
- Decompression on write to backing store
- On average a compression ratio of 4:1 is achieved

Backing Store Compression

- Quartz Compositor can composite directly from compressed data
- System level optimization
    - No code changes required

Velocity Engine

- Some graphics operations make use of the Velocity Engine
    - Software Quartz Compositor
    - Blitters, e.g., QD **CopyBits** to window backing store
    - Backing store compression
- Recommendation
    - If you allocate bitmaps, make **rowBytes** divisible by 16

Demo

Quartz Performance

Ralph Brunner

QuickDraw Performance on Mac OS X

Joseph Maurer
QuickDraw Bavarian

QuickDraw Performance

- Benchmarks and the fps myth
- What else is different?
- Recommendations and special cases
- Demos

Benchmarks and the fps Myth

- We broke benchmarks!
- Use Sampling, MicroSeconds
- How many fps do you need?
- Reduce flushing

H/W Acceleration

- H/W acceleration is different on Mac OS X
- Quartz Exteme is accelerated
- QuickDraw and Quartz 2D are not H/W accelerated

Dirty Region Maintenance

- Region processing is costly
- Non-trivial clip regions
- Recommendation:
    - Use **QDSetDirtyRgn(port, bigRectRgn)**!

      (and not **QDAddRectToDirtyRgn**)

LockPortBits/UnlockPortBits

- Recommendation:

```
LockPortBits(GetWindowPort(window))
//  .. your QD drawing sequence ...
UnlockPortBits();
```

- Nested calls are “free”
- But be careful!

QuickDraw Performance Demo

- Scrolling
- CopyBits throughput
- 88000 lines

Demo

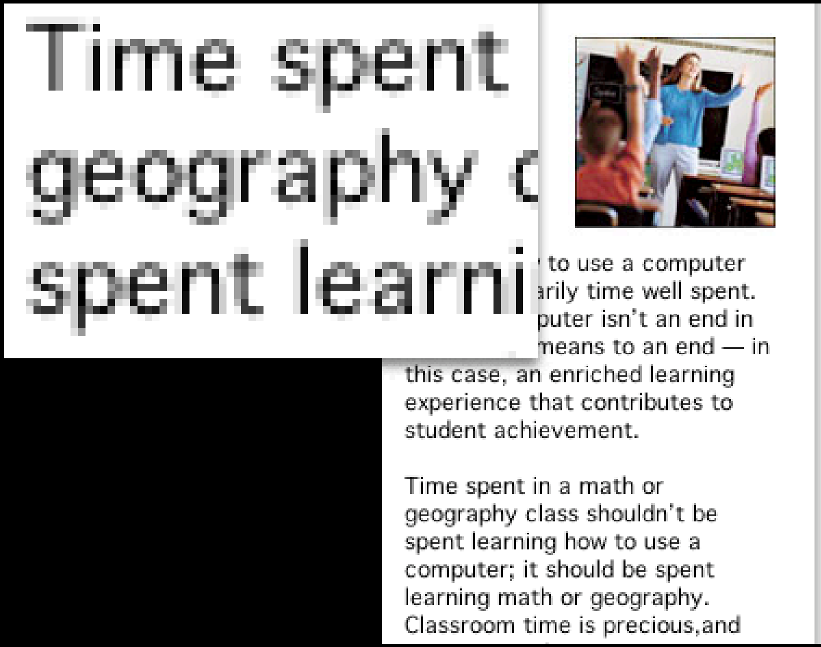
QuickDraw Performance on Mac OS X

Joseph Maurer

Tips and Tricks

Haroon Sheikh

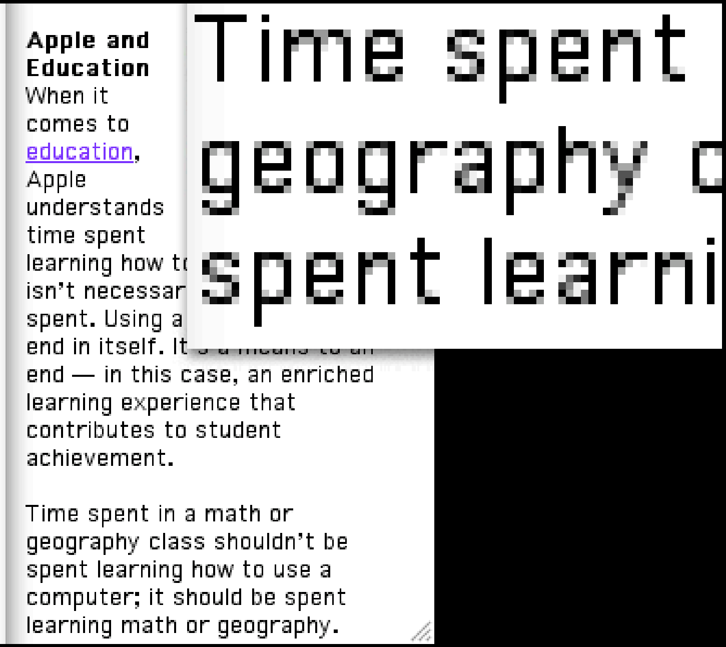
Text Anti-aliasing



Time spent
geography c
spent learni

to use a computer
arily time well spent.
puter isn't an end in
means to an end — in
this case, an enriched learning
experience that contributes to
student achievement.

Time spent in a math or
geography class shouldn't be
spent learning how to use a
computer; it should be spent
learning math or geography.
Classroom time is precious,and



Apple and Education
When it
comes to
education,
Apple
understands
time spent
learning how to
isn't necessar
spent. Using a
end in itself. It
end — in this case, an enriched
learning experience that
contributes to student
achievement.

Time spent in a math or
geography class shouldn't be
spent learning how to use a
computer; it should be spent
learning math or geography.

Time spent
geography c
spent learni

Quartz

QuickDraw Text

Using Quartz Text Anti-aliasing In QuickDraw

- Not on by default
- Application wide: **QDSwapTextFlags**
  - **kQDUseCGTextRendering**
    - Uses CG to render text, but metrics from QD
  - **kQDUseCGTextMetrics**
    - Use CG to render text and uses CG metrics
  - **kQDUseTrueTypeScalerGlyphs**
    - Use QuickDraw traditional rendering
- Port based: **QDSwapPortTextFlags**

QD Text

- Limitations
  - Not all styles and transfer modes are supported
  - Glyph “squishing” not available
  - Using CG Text metrics will cause re-layout
  - Using CG text metrics can be slower because of sub-pixel positioning
- Q&A available

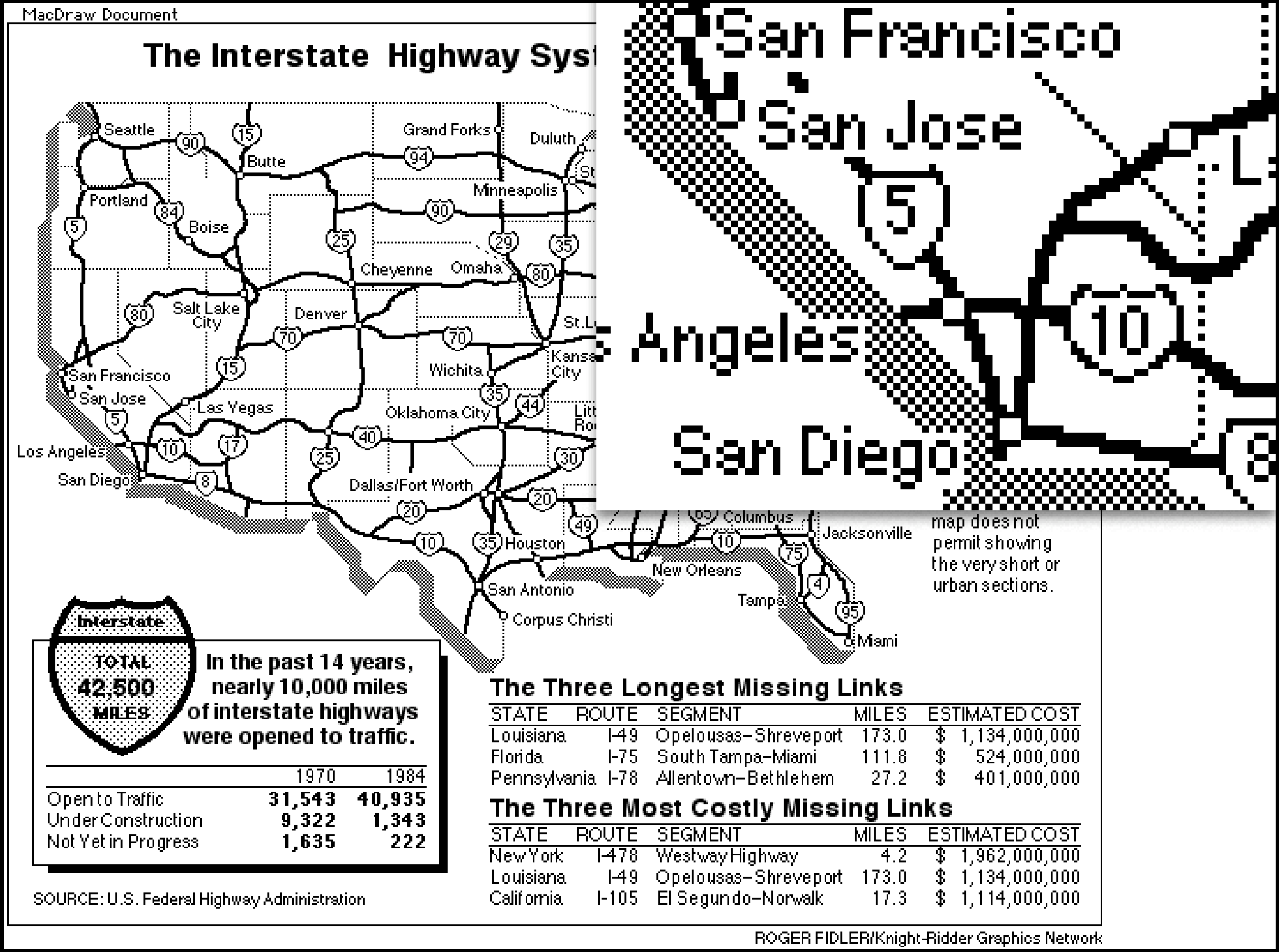
Using Quartz 2D In Carbon Applications

- Getting a CGContext:
  QDBeginCGContext
- To go back to QD drawing
  QDEndCGContext
- Tip: Make sure you flush the context
- Tip: Use **CGContextSynchronize** to synchronize flushes from multiple contexts
- Tip: Replace PS Picture comments with Quartz 2D

Drawing PICTs Using Quartz

- **QDPictDrawToCGContext**
  - Uses CG rendering to draw the PICTs
  - Substitutes shades of gray for pixmap patterns
  - Respects Quartz 2D transformations

QuickDraw PICT Rendering



The Interstate Highway Syst

In the past 14 years, nearly 10,000 miles of interstate highways were opened to traffic.

	1970	1984
Open to Traffic	31,543	40,935
Under Construction	9,322	1,343
Not Yet in Progress	1,635	222

map does not permit showing the very short or urban sections.

The Three Longest Missing Links

STATE	ROUTE	SEGMENT	MILES	ESTIMATED COST
Louisiana	I-49	Opelousas–Shreveport	173.0	$ 1,134,000,000
Florida	I-75	South Tampa–Miami	111.8	$ 524,000,000
Pennsylvania	I-78	Allentown–Bethlehem	27.2	$ 401,000,000

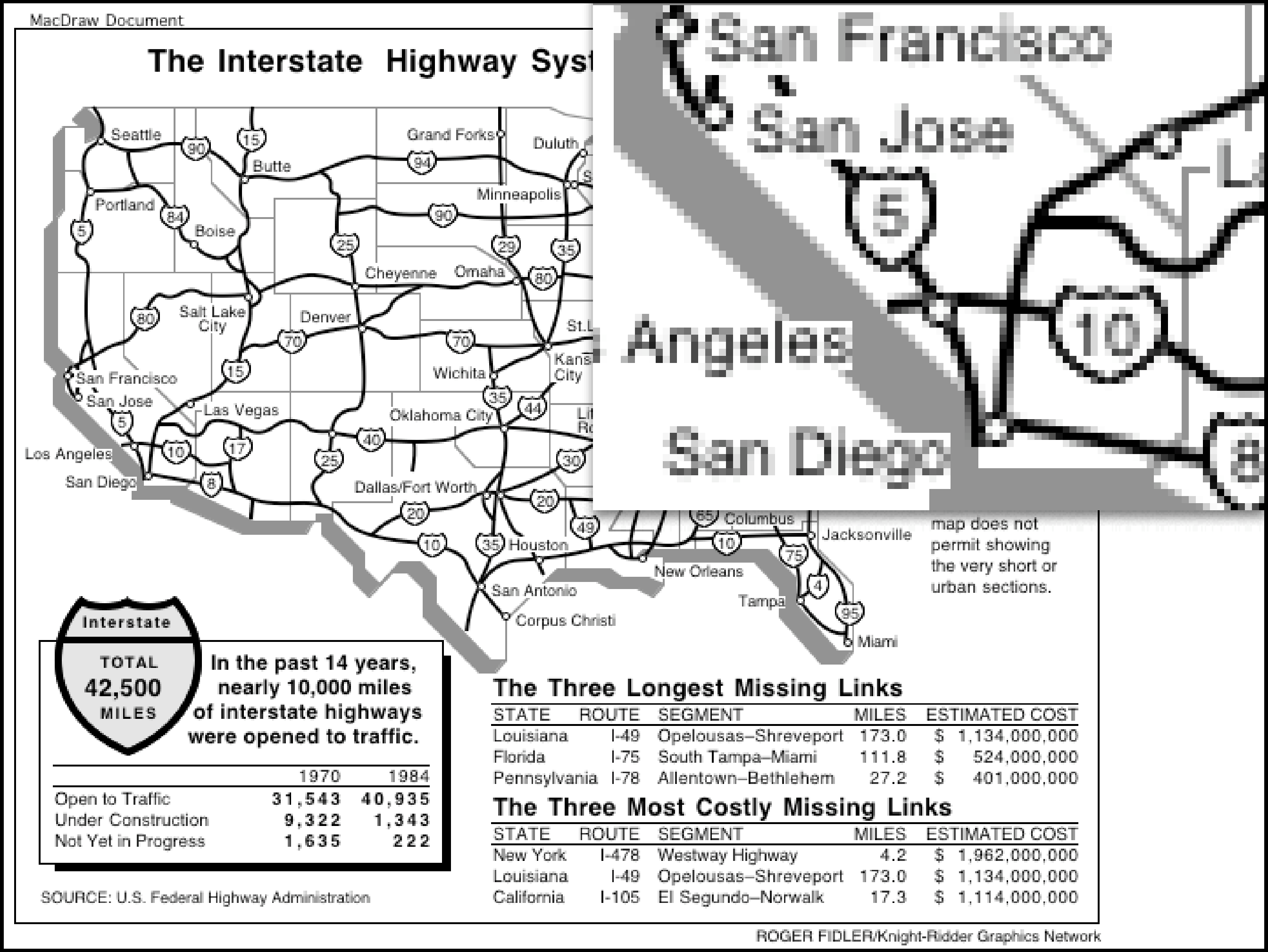
The Three Most Costly Missing Links

STATE	ROUTE	SEGMENT	MILES	ESTIMATED COST
New York	I-478	Westway Highway	4.2	$ 1,962,000,000
Louisiana	I-49	Opelousas–Shreveport	173.0	$ 1,134,000,000
California	I-105	El Segundo–Norwalk	17.3	$ 1,114,000,000

SOURCE: U.S. Federal Highway Administration

ROGER FIDLER/Knight-Ridder Graphics Network

PICT Drawn With Quartz 2D



The Interstate Highway Syst

map does not permit showing the very short or urban sections.

Interstate TOTAL 42,500 MILES

In the past 14 years, nearly 10,000 miles of interstate highways were opened to traffic.

	1970	1984
Open to Traffic	31,543	40,935
Under Construction	9,322	1,343
Not Yet in Progress	1,635	222

SOURCE: U.S. Federal Highway Administration

The Three Longest Missing Links

STATE	ROUTE	SEGMENT	MILES	ESTIMATED COST
Louisiana	I-49	Opelousas–Shreveport	173.0	$ 1,134,000,000
Florida	I-75	South Tampa–Miami	111.8	$ 524,000,000
Pennsylvania	I-78	Allentown–Bethlehem	27.2	$ 401,000,000

The Three Most Costly Missing Links

STATE	ROUTE	SEGMENT	MILES	ESTIMATED COST
New York	I-478	Westway Highway	4.2	$ 1,962,000,000
Louisiana	I-49	Opelousas–Shreveport	173.0	$ 1,134,000,000
California	I-105	El Segundo–Norwalk	17.3	$ 1,114,000,000

ROGER FIDLER/Knight-Ridder Graphics Network

QuickDraw PICT Rendering

- Limitations
    - No special raster ops
- Performance difference
    - Conversion of QD primitives to Quartz
    - Tip: Convert PICTs to PDFs

Quartz 2D and QuickTime

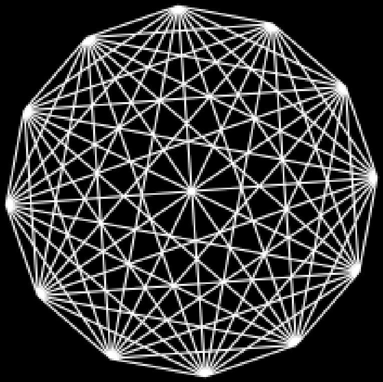
- Displaying an image using QuickTime and Quartz 2D
    - Create a 32bit ARGB **GWorld**
    - Draw into **GWorld** using QT
    - Get Color Profile from QT
    - Convert to **CGColorSpace**
    - Create a **CGImageRef**
    - Draw using **CGContextDrawImage**

Working With Images

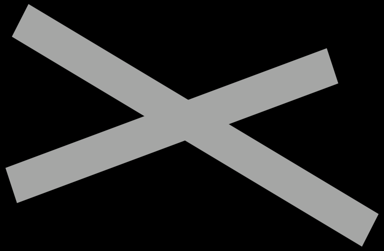
- **CGImageRefs** are not cached
    - Tip: cache down-sampled or colormatched results
- Tip: Use JPEG/PNG data providers
- Tip: Use custom data providers for non-native formats

Complicated Paths

- Single complicated path vs. multiple simple line based paths
    - Difference is in the region of intersection



Multiple Lines



Single Path

CGDirectDisplay

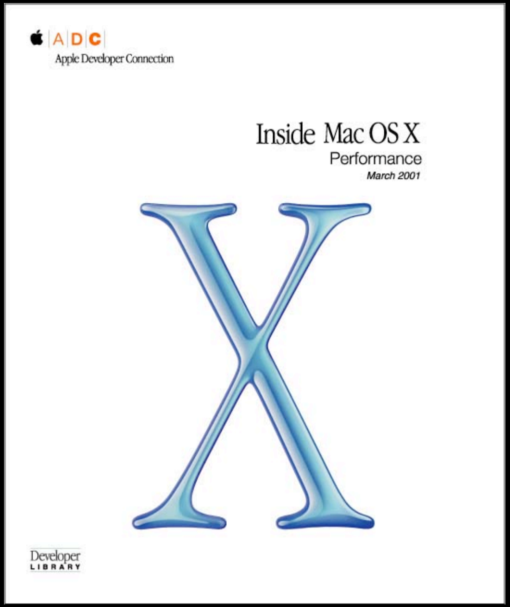
- New API to list all online Displays
- New API for reconfiguring displays
- New API for querying display properties
    - Check if Quartz Extreme is running
- API to get screen DPI

Development Tools

- /Developer/Application/QuartzDebug
- /Developer/Application/Sampler
- /usr/bin/sample command line tool

Documentation

Graphics and Imaging Performance


ADC
Apple Developer Connection

Inside Mac OS X
Performance
March 2001

Developer
LIBRARY

- Performance
- Drawing With Quartz 2D
- Quartz 2D Reference
- Quartz Primer

Documentation > Documentation Essentials
developer.apple.com/techpubs/macosx/Essentials/devessentials.html
Documentation > Core Technologies > Graphics > Quartz 2D
developer.apple.com/techpubs/macosx/CoreTechnologies/graphics/Quartz2D/quartz2d.html

More Documentation

Graphics and Imaging Performance

- TechNote: QuickDraw Performance
- QA: QuickDraw Text Anti-aliasing using Quartz

Roadmap

500 Graphics and Imaging Overview	Room A2 **Tue., 10:30am**
501 Quartz 2D and PDF	Room A2 **Tue., 2:00pm**
503 Exploring the Quartz Compositor	Hall 2 **Tue., 3:30pm**
504 OpenGL Graphics Programmability:	Room A2 **Tue., 5:00pm**

Roadmap

505 OpenGL Integrated Graphics I	Room J **Wed., 9:00am**
506 OpenGL Integrated Graphics II	Room J **Wed., 10:30am**
109 Darwin Printing	Room J **Wed., 2:00pm**
509 ColorSync and Digital Media	Room C **Wed., 5:00pm**

Roadmap

510 Printing and Mac OS X	Hall 2 **Thurs., 10:30am**
513 OpenGL Advanced 3D	Room J **Thurs., 3:30pm**
514 OpenGL: Performance and Optimization	Room J **Thurs., 5:00pm**
515 Image Capture Framework	Room C **Fri., 2:00pm**

Roadmap

516 Graphics and Imaging Performance Tuning	Hall 2 **Fri., 3:30pm**
FF018 Graphics and Imaging	Room J1 **Fri., 5:00pm**

Who to Contact

Travis Brown
Graphics and Imaging Evangelist
travis@apple.com

Q&A

Travis Brown
Graphics and Imaging Evangelist
Worldwide Developer Relations

http://developer.apple.com/wwdc2002/urls.html

WWDC2002

WWDC2002

WWDC2002